LAPORAN MINGGU 8

Praktikum Fundamental Programming Structures In Java

Teknik Pemograman

Tugas ini untuk memenuhi Mata Kuliah Teknik Pemograman Praktek

Disusun Oleh :

Reka Briyan Cahya Heryana – 211524024

PROGRAM STUDI D4 TEKNIK INFOMATIKA

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

POLITEKNIK NEGERI BANDUNG

2022

TASK 1.1 – 1.3

1. Hasil Program

```
run:
Cylinder: radius=1.0 height=1.0 base area=12.566370614359172 volume=12.566370614359172
Cylinder : subclass of Circle[radius=1.0 color=red]hight=1.0

Cylinder: radius=1.0 height=10.0 base area=69.11503837897544 volume=691.1503837897544
Cylinder : subclass of Circle[radius=1.0 color=red]hight=10.0

Cylinder: radius=2.0 height=10.0 base area=150.79644737231007 volume=1507.9644737231006
Cylinder : subclass of Circle[radius=2.0 color=red]hight=10.0

BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Jawaban dari Soal yang dipertanyakan

Circle.java

```
public class Circle { // Save as "Circle.java"
 // private instance variable, not accessible from outside this class
    private double radius;
    private String color;

public String getColor(){
    return color;
}

public void setColor(String color){
    this.color = color;
}

 // Constructors (overloaded)
 /** Constructs a Circle instance with default value for radius and color */
 public Circle() { // 1st (default) constructor
    radius = 1.0;
    color = "red";
 }

 /** Constructs a Circle instance with the given radius and default color */
 public Circle(double r) { // 2nd constructor
 radius = r;
 color = "red";
 }
```

Cylinder.java

```
public class Cylinder extends Circle { // Save as "Cylinder.java"
 private double height; // private variable

 // Constructor with default color, radius and height
 public Cylinder() {
 super(); // call superclass no-arg constructor Circle()
 height = 1.0;
 }
 // Constructor with default radius, color but given height
 public Cylinder(double height) {
 super(); // call superclass no-arg constructor Circle()
 this.height = height;
 }
 // Constructor with default color, but given radius, height
 public Cylinder(double radius, double height) {
 super(radius); // call superclass constructor Circle(r)
 this.height = height;
 }

 // A public method for retrieving the height
 public double getHeight() {
 return height;
 }
```

TestCylinder.java

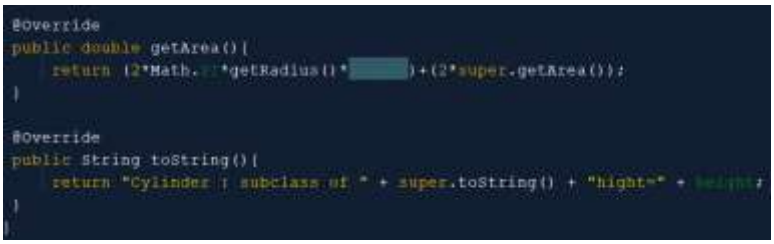
```
public class TestCylinder { // save as "TestCylinder.java"
 public static void main (String[] args) {
 // Declare and allocate a new instance of cylinder
 // with default color, radius, and height
 Cylinder c1 = new Cylinder();
 System.out.println("Cylinder:"
 + " radius=" + c1.getRadius()
 + " height=" + c1.getHeight()
 + " base area=" + c1.getArea()
 + " volume=" + c1.getVolume()
 + "\n" + c1.tostring() + "\n");

 // Declare and allocate a new instance of cylinder
 // specifying height, with default color and radius
 Cylinder c2 = new Cylinder(10.0);
 System.out.println("Cylinder:"
 + " radius=" + c2.getRadius()
 + " height=" + c2.getHeight()
 + " base area=" + c2.getArea()
 + " volume=" + c2.getVolume()
 + "\n" + c2.tostring()+ "\n");

 // Declare and allocate a new instance of cylinder
 // specifying radius and height, with default color
 Cylinder c3 = new Cylinder(2.0, 10.0);
```

3. Masalah yang dihadapi
   - Task 1.1
     Saya memberikan variabel color bertipe string, lalu memberikan 3 kontruktor pada kelas circle serta saya menambahkan getter dan setter.
   - Task 1.2
     Saya menggunakan ovveride di calss cylinder dan TestCylinder (Main). Overriding ini merupakan sebuah fitur yang dimana sebuah subkelas atau anak kelas yang menyediakan sebuah implementasi yang jelas dari motode yang sudah tersedia oleh salah satu dari super kelas atau parent kelas.
   - Task 1.3
     Menambahkan metode tostring pada kelas TestCylinder (Main). Method ini digunakan untuk merepresentasikan sebuah objek, kedalam tipe String. Jika kita inign menampilkan sebuah objek secara implisit dan sebenarnya compiler sudah memanggil method toString(). Untuk memunculkan ke layar gunakan toString().

4. Solusi dari Masalah yang Dihadapi
   - Task 1.1

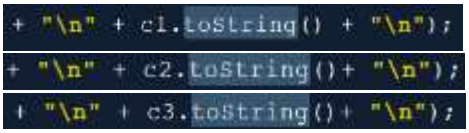
```
public String getColor(){
    return color;
}

public void setColor(String color){
    this.color = color;
}
```

- Task 1.2



```
@Override
public double getArea(){
    return (2*Math.PI*getRadius()*height)+(2*super.getArea());
}

@Override
public String toString(){
    return "Cylinder : subclass of " + super.toString() + "hight=" + height;
}
}
```

- Task 1.3



```
+ "\n" + c1.toString() + "\n");
+ "\n" + c2.toString()+ "\n");
+ "\n" + c3.toString()+ "\n");
```

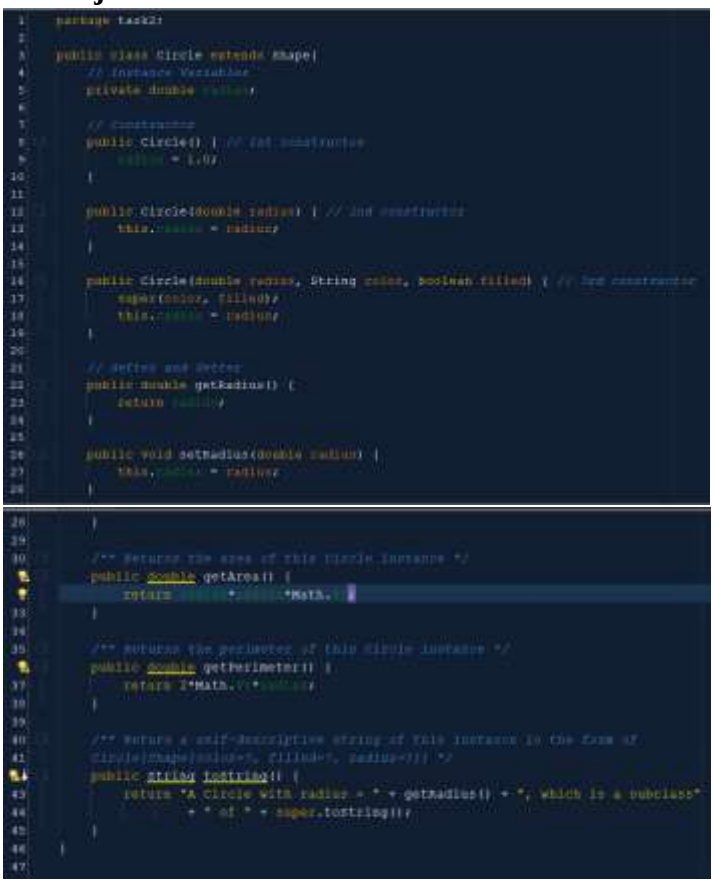
5. Teman yang Membantu
   - Tidak ada

Task 2.1

1. Hasil Program

2. Jawaban dari Soal yang Dipertanyakan

Circle.java

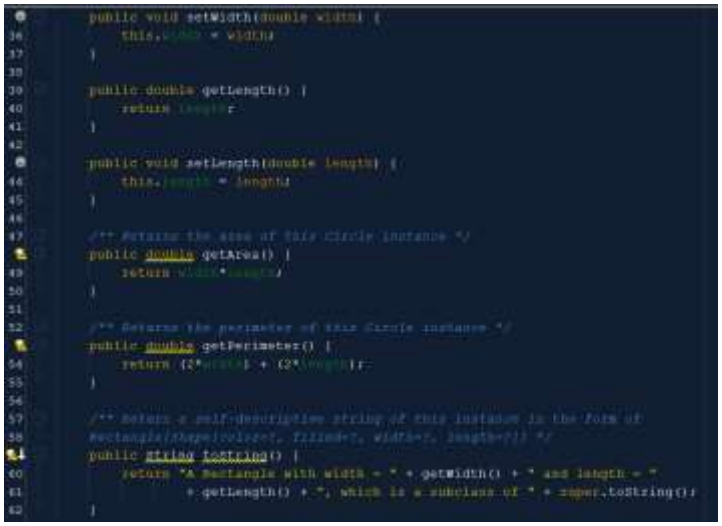


Shape.java

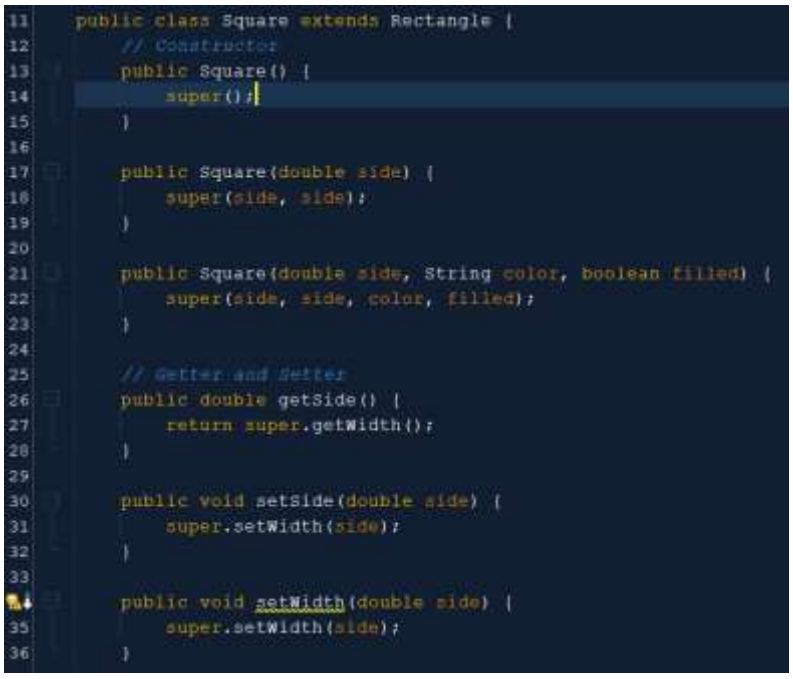
```
public class Shape {
    // Instance Variables
    private String color;
    private boolean filled;

    // Constructors
    public Shape() { // 1st constructor
        color = "green";
        filled = true;
    }

    public Shape(String color, boolean filled) { // 2nd constructor
        this.color = color;
        this.filled = filled;
    }

    // Getter and Setter
    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public boolean isFilled() {
        return filled;
    }

    public void setFilled(boolean filled) {
        this.filled = filled;
    }

    /** Return a self-descriptive string of this instance in the form of
    Circle[radius=?,color=?] */
    public String toString() {
        if(isFilled()) {
            return "A Shape with color of " + getColor() + " and filled";
        }
        else {
            return "A Shape with color of " + getColor() + " and not filled";
        }
    }
}
```

Rectangle.java

```
public class Rectangle extends Shape {
    // Instance Variables
    private double width;
    private double length;

    // Constructor
    public Rectangle() {
        width = 1.0;
        length = 1.0;
    }

    public Rectangle(double width, double length) {
        this.width = width;
        this.length = length;
    }

    public Rectangle(double width, double length, String color, boolean filled) {
        super(color, filled);
        this.width = width;
        this.length = length;
    }

    // Getter and Setter
    public double getWidth() {
        return width;
    }
```



Square.java



TestShape.java

```
public class TestShape {
    public static void main(String[] args) {
        // Declare and allocate a new instance of Shape
        // with default color and filled
        Shape s1 = new Shape();
        System.out.println(s1.toString());

        // Declare and allocate a new instance of Shape
        // with value color and filled
        Shape s2 = new Shape("red", false);
        System.out.println(s2.toString());

        // Declare and allocate a new instance of Circle
        // with value radius, color, and not filled
        Circle c1 = new Circle(7.0, "red", false);
        System.out.println(c1.toString());
        System.out.println("Luas = " + c1.getArea());
        System.out.println("Keliling = " + c1.getPerimeter() + "\n");

        // Declare and allocate a new instance of Circle
        // with value width, height, color, and not filled
        Rectangle r1 = new Rectangle(3.0, 5.0, "blue", true);
        System.out.println(r1.toString());
        System.out.println("Luas = " + r1.getArea());
        System.out.println("Keliling = " + r1.getPerimeter() + "\n");
```



```
        // Declare and allocate a new instance of Circle
        // with value width, height, color, and not filled
        Square sq1 = new Square(3.0, "yellow", true);
        System.out.println(sq1.toString());
        System.out.println("Luas = " + sq1.getArea());
        System.out.println("Keliling = " + sq1.getPerimeter());
    }
```

3. Masalah yang Dihadapi
   - Tulis dua subclass dari shape yang disebut circle dan rectangle
   - Tulis kelas square sebagai subclass dari rectangle
   - Square dapat dimodelkan sebagai sebclass dari rectangle. Kotak tidak memiliki instance variabel, namun mewarisi lebar dan panjang variabel dari superclassnya Rectangle
4. Solusi dari Masalah yang Dihadapi
   - Kasus ini tidak berbeda jauh dengan task 1 yang memerlukan 3 konstruktor dan menggunakan setter dan getter
5. Teman yang Membantu
   - Tidak ada

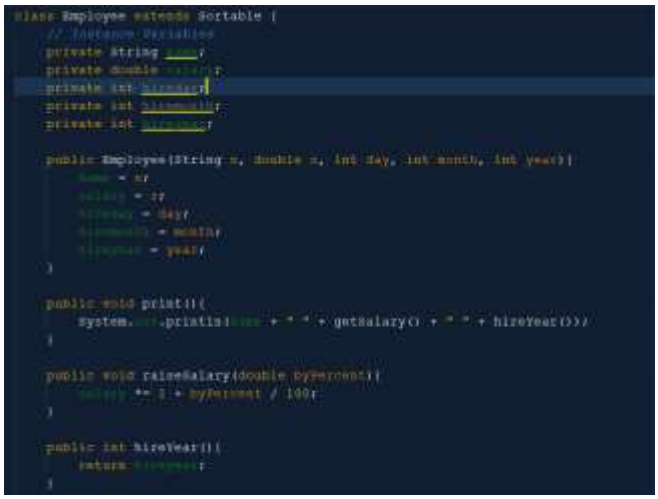
Task 3.1 – 3.2

1. Hasil Program

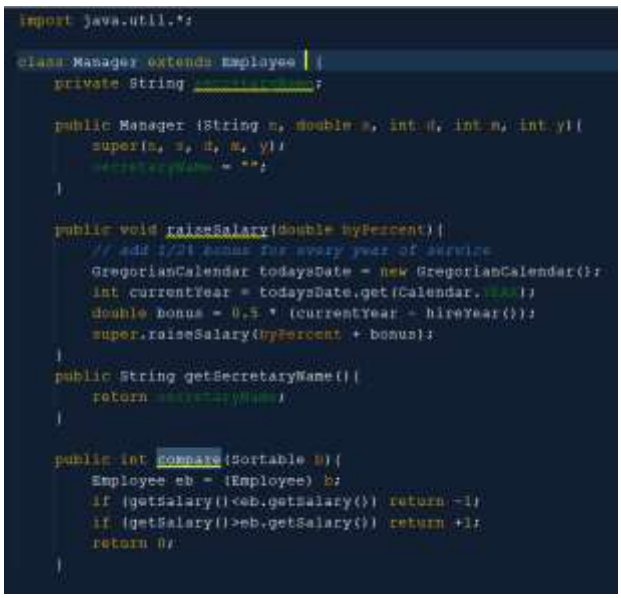
```
LIST :
Antonio Rossi 2100000.0 1989
Maria Bianchi 3012500.0 1991
Isabel Vidal 3150000.0 1993
SORTED LIST :
Isabel Vidal 3307500.0 1993
Maria Bianchi 3630062.5 1991
Antonio Rossi 2205000.0 1989
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Jawaban dari Soal yang Dipertanyakan

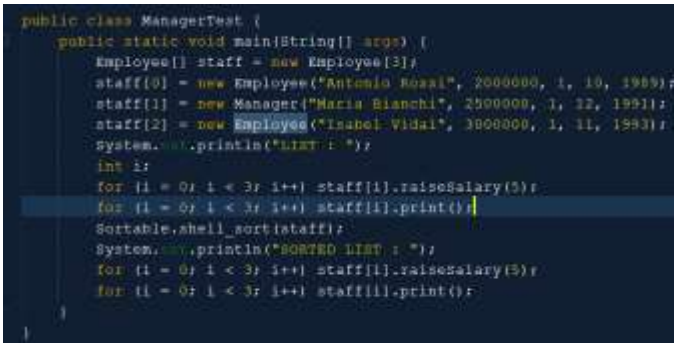
Employee.java



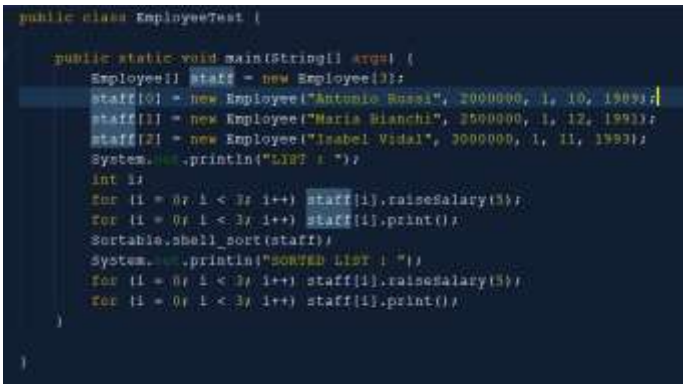
Manager.java



ManagerTest.java



Employee.java



Sortable.java

3. Masalah yang Dihadapi
   - Tidak mengetahui dan baru pertama kali mendengar shortshell
4. Solusi dari Masalah yang Dihadapi
   - Mencari arti dan memahami konsep serta memodifikasi sesuai dengan kasus ini
5. Teman yang Membantu
   - Tidak ada